Online Estimation of Trifocal Tensors for Augmenting Live Video

Jia Li, Robert Laganière
University of Ottawa
School of Information Technology and Engineering
Ottawa, K1N 6N5, Canada
jiali,laganier@site.uottawa.ca

Gerhard Roth
National Research Council
NRC Institute for Information Technology
Ottawa, K1A 0R6, Canada
Gerhard.Roth@nrc-cnrc.gc.ca

Abstract

We propose a method to augment live video based on the tracking of natural features, and the online estimation of the trinocular geometry. Previous without-marker approaches require the computation of camera pose to render virtual objects. The strength of our proposed method is that it doesn't require tracking of camera pose, and exploits the usual advantages of marker-based approaches for a fast implementation. A 3-view AR system is used to demonstrate our approach. It consists of an uncalibrated camera that moves freely inside the scene of interest, and of three reference frames taken at the time of system initialization. As the camera is moving, image features taken from an initial triplet set are tracked throughout the video sequence. And the trifocal tensor associated with each frame is estimated online. With this tensor, the square pattern that was visible in the reference frames is transferred to the video. This invisible pattern is then used by the ARToolkit to embed virtual objects.

1. Introduction

Existing augmented reality approaches relying on special markers have achieved impressive results in video-based applications. The markers may be patterns or landmarks that are introduced in the scene, or they may be naturally occurring features selected in the scene. The location of these markers in each video frame then define a 3D virtual coordinate system. The transformation of the virtual object is then computed with respect to these coordinate systems and thus inserted into each video frame. The ARToolkit [1, 10] works by tracking a square marker pattern containing at least four coplanar corners so that the 3D-to-2D transformation is represented by a plane-to-plane homography. In Kutulakos' calibration-free system [8], four or more non-coplanar points are tracked along the video and an affine object representation is used to overlay virtual objects on an orthographic video stream without camera calibration. The greatest advantage of these approaches that use markers or control points is their capacity to operate at frame rate. It comes at the expense of requiring that all markers be visible in every frame. Therefore, these methods do not offer robustness to occlusion of any markers, which restricts the range of views in which augmentations can take place.

Approaches based on the tracking of natural features provide a general solution to augment video, especially when a scene can not be prepared. With tracked features, the technology of structure-and-motion (SaM) is involved in recovering camera motion and the scene's 3D structure, which are then used to incorporate virtual objects into the scene [21, 4]. Inside accurately reconstructed scenes, occlusions of real and virtual objects can be properly handled. The disadvantages of this approach are: (1) Camera calibration is required in order to obtain metric structure and motion; (2) Computational requirements to deal with live videos exceeds the capability of realtime processing systems. Therefore, these approaches are only suitable for off-line video processing.

To calculate camera pose and scene structure, feature correspondences need to be established throughout the sequence. The approach used in [21] is to first match features between all consecutive pairs of video frames. Correspondences over many frames are produced by merging the matches of overlapping frames. Then a frame-to-frame transformation is computed and the camera pose related to every frame is the product of all rotations and translations from previous frames. The main disadvantage of this recursive approach is that it is easily corrupted by accumulated error, especially in long sequences.

Another approach to establishing correspondences consists in registering every video frame with respect to some reference images of the scene, which are captured during an initialization phase. Since feature matching is always performed between the current frame and reference images, the geometrical transformations, including homographic, epipolar, projective or trifocal transformations, are computed independently in each frame. Therefore, error accumulation is avoided. A possible approach to using keyframes for augmented reality is to calculate the 3D structure of features matched in the keyframes, then track these features in every other frame in order to obtain 2D-3D matches to compute projection matrices [4]. Unfortunately, the computational complexity of such reconstruction and auto-calibration approaches hinders their application to online processing. The latest progress in using multiple keyframes for the recovery of camera pose and the tracking and augmenting of a known object is presented in [17]. However, a 3D CAD model of the target object is required a priori.

In this paper we propose an approach to the online augmentation of live video that doesn't require camera pose information or camera internal parameters. It combines the strengths of both keyframe-based techniques and the ARToolkit to achieve a good performance in terms of robustness and speed. In the off-line initialization stage, three reference images are captured from different viewpoints with a square pattern temporarily placed in the scene. Then, the pattern is withdrawn, and as the camera is moving freely inside the scene, image features taken from an initial set of corresponding triplets detected on the three reference images are tracked across the video sequence. The trifocal tensor associated with each frame and two of the reference images is then estimated in realtime. Using these computed tensors, the square pattern, which was visible in the reference images, can be transferred to the moving frames. Virtual objects are then placed onto the video, by feeding this virtual pattern to the ARToolkit.

The method proposed in this paper uses a 3-step paradigm where (1) feature points are tracked, (2) a tensor is estimated, and (3) additional points and virtual pattern are transferred. An important aspect of this work is the fact that the updated tensors associated with every moving frame and the two fixed reference views are computed based on a common set of detected features. As a consequence, there is no drift problem. At the same time, the viewpoints will always remain sufficiently wide to ensure an accurate estimation of the tensor.

This method also distinguishes itself in three ways: (1) Random sampling techniques are avoided in the process of tensor estimation because of their complexity. Instead, the tensor is estimated by using an algebraic minimization method and its accuracy is improved afterwards through a quick removal of outliers. (2) A simple tracker is used to provide an evolving set of point triplets. This set is also updated by recovering lost points and correcting mismatches using trifocal transfer. Stable performance of tracking over a long video sequence is also ensured by automatically resetting the tracker when the size of the set of tracked points becomes too small. (3) The method is flexible in the sense that for simple applications, such as video labeling or notation, the line segments and vertices of the rendered virtual objects may be transferred from the fixed reference views to the moving camera view using the trifocal tensor that we compute at each frame. For a rudimentary polyhedral object, this transfer process is quick and direct. Even for a complex object, the transfer of its rendered triangulation is still straightforward as long as its rendering in the two fixed reference images is available.

It is also important to note that the proposed 3-view system can easily be extended to the case of multiple views. An additional first step would be to identify the two closest reference views with which tensor estimation is to be performed. This way the scope of this vision system could be scaled to a size required by a given application.

The rest of this paper is organized as follows. Section 2 gives some preliminaries and notations on trifocal tensor. Section 3 gives an outline of the approach. Section 4 discusses the tensor estimation process. Section 5 describes updating of the set of points to be tracked. Section 6 deals with inserting virtual objects into live video. Finally, Section 8 contains the conclusion.

2. Preliminaries

The trifocal tensor describes the projective geometric relations of image triplets taken from cameras [7]. If the camera matrix of the first view is in canonical form, $\mathbf{P_1} = [\mathbf{I}|\mathbf{0}]$, and the camera matrices of the other two views are expressed as $\mathbf{P_2} = [\mathbf{A}|\mathbf{e}']$, $\mathbf{P_3} = [\mathbf{B}|\mathbf{e}'']$, where $\mathbf{A}$ and $\mathbf{B}$ are 3×3 matrices, and, $\mathbf{e}'$ and $\mathbf{e}''$ are the epipoles corresponding to the image of the center of the first camera on the image plane of the second and third cameras respectively, then the $3 \times 3 \times 3$ trifocal tensor could be denoted as $\mathbf{T} = [\mathbf{T_1}, \mathbf{T_2}, \mathbf{T_3}]^T$, with:

$$\mathbf{T_i} = \mathbf{a_i}\mathbf{e}''^T - \mathbf{e}'\mathbf{b_i}^T \tag{1}$$

It is known that this tensor provides a more accurate and stable description of three views' geometry than the fundamental matrices between each pair of views. The most attractive characteristic of the trifocal tensor is the transfer of points and lines, i.e. a point/line in one image can be computed from its correspondence in the other two images. If $(\mathbf{l}, \mathbf{l}', \mathbf{l}'')$ is a set of corresponding lines and $(\mathbf{x}, \mathbf{x}', \mathbf{x}'')$ is a set of corresponding points in three images, the transfer operations can be represented by following equations:

$$\mathbf{l^T} = \mathbf{l'^T}[\mathbf{T_1}, \mathbf{T_2}, \mathbf{T_3}]\mathbf{l}'' \tag{2}$$

$$[\mathbf{x}']_\mathbf{x}(\sum_\mathbf{i} \mathbf{x^i}\mathbf{T_i})[\mathbf{x}'']_\mathbf{x} = \mathbf{0} \tag{3}$$

The trifocal tensor can be estimated from image correspondences alone without knowledge of the camera parameters. This means that no explicit 3D information is required in order to work with tensors. Moreover, the normalized projection matrices of three cameras corresponding to a tensor may be chosen as

$$\mathbf{P_1} = [\mathbf{I}|\mathbf{0}] \tag{4}$$

$$\mathbf{P_2} = [[\mathbf{T_1}, \mathbf{T_2}, \mathbf{T_3}]\mathrm{e}''|\mathrm{e}'] \tag{5}$$

$$\mathbf{P_3} = [(\mathrm{e}''\mathrm{e}''^T - I)[\mathbf{T_1}^T, \mathbf{T_2}^T, \mathbf{T_3}^T]\mathrm{e}'|\mathrm{e}''] \tag{6}$$

3. Outline of the Approach

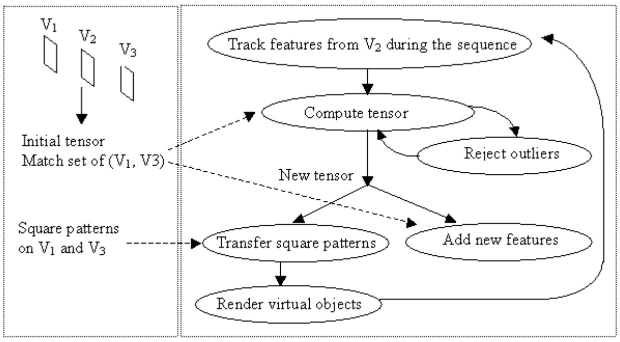


Figure 1. Flow chart of the proposed approach

Our proposed approach is illustrated in Figure 1. Figure 2 shows a schematic overview of the 3-step procedure. The system has, as input, three camera views, denoted by $\mathbf{V}_1$, $\mathbf{V}_2$, $\mathbf{V}_3$ respectively. They contain a square pattern which is purposely placed inside the scene at the capture time. Note that this pattern does not have to be present anymore once the three keyframes are obtained.

The initialization step consists in obtaining both an initial estimate of the tensor and a large set of matched triplets. Several alternatives can be envisaged in order to achieve this goal, including a tensor-based guided-matching [20] and the PVT tool described in [14]. The feature points of the obtained triplet set that belong to one reference view will constitute the initial set of point to be tracked. Match pairs between the other fixed views will serve as a match pool that will be used, during the process, to update the list of points to be tracked.

Once the initialization process is completed, the online tensor estimation and augmentation process can start. The detected points in one reference view are tracked from one frame to the next. This leads to new positions of the points for which we still have the correspondences in the two fixed ones. Using this updated triplet set, robust and fast estimation of the tensor is achieved; this aspect is discussed in detail in Section 4. Once a new tensor is obtained, the square pattern specified in the two fixed reference views, $\mathbf{V}_1$ and $\mathbf{V}_3$, is transferred into the moving camera view to generate a virtual image of this pattern, with which the ARToolKit method is implemented to embed the virtual object.

Obviously, when points are tracked over time, more and more features are unavoidably lost. And if nothing is done, the tracked set will eventually vanish. To overcome this problem, the match set is updated after each tensor estimation. Indeed, using the pool of match pair available in the two fixed views, it becomes possible to transfer new points on the image using the newly estimated tensor. This last step ensures the long term viability of the estimation process. In a multi-camera implementation, points from view close to the current reference views would also be transferred, thus allowing the identification of the view toward which the moving camera is transiting.

4. Online Estimation of the Trifocal Tensor

The properties of the trifocal tensor have been exploited in various video-based applications, including camera pose estimation [14], object-based video compression [15], 3D modelling [2] and augmented reality [21]. All these works propose an off-line processing of recorded videos and are based on a similar framework (as mentioned in Section 1) that can be summarized as follows. Cross-correlation and guided matching based on fundamental matrices are used to find matches between two adjacent frames. The resulting overlapping match sets are then used to build putative triplet sets of correspondences between three consecutive frames, from which the tensor of this image triplet is computed. Finally, all tensors of the sequence are linked together into a long chain. A random sampling technique is used to estimate the tensor and fundamental matrix. Again, this mechanism suffer error accumulation. And the accuracy of the resulting tensor is doubtful because of the small motions involved in consecutive frames.

In this section we describe a method to estimate the trifocal tensor based on the utilization of three keyframes. Its capability to implement in realtime has been proven by experiments.

4.1. Overview of Estimation Methods

The trifocal tensor is computed from image correspondences of points and evaluated in terms of residual error between transferred points and measurements. Existing estimation methods could be roughly divided into two classes. The first class, that includes linear least-square solution, algebraic minimization and geometric distance minimization

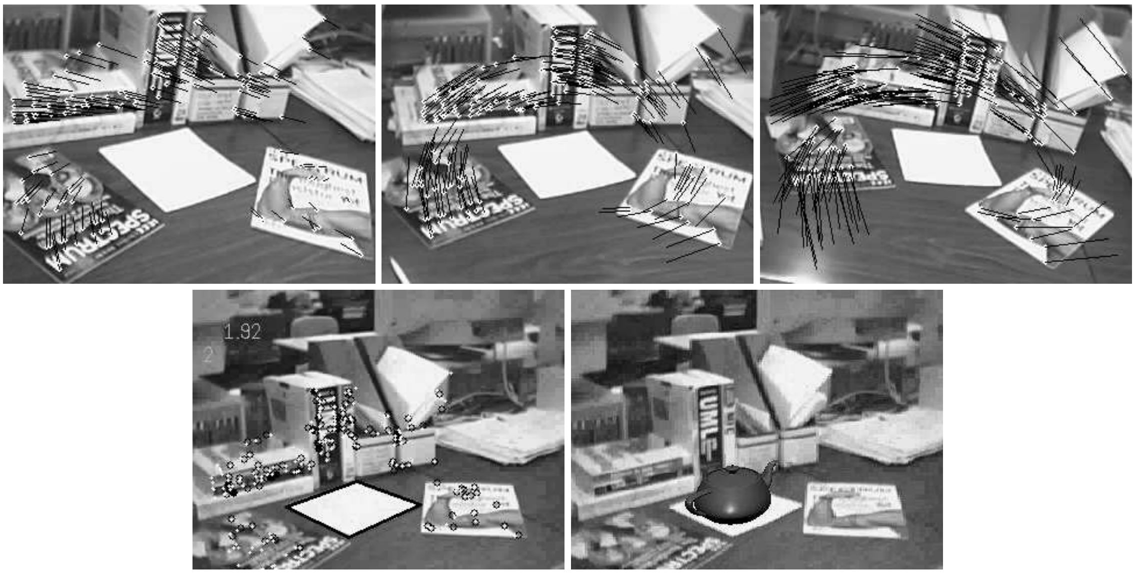


Figure 2. Three reference images,V_1, V_2, V_3, are shown in the top column.Matched corners from the initial set of triplets are shown superimposed on each image. A blank paper was selected on the reference images by manually selecting its four corners at system initialization time. For each video frame, the bottom-left image, the trifocal tensor is estimated from tracked corners (blank circles). All matches of the view pair (V_1,V_3) can be transferred to this frame. The transferred points (white dots) are used for updating the tracker's point set. Transfer of the four corners on the blank paper gives a virtual plane location, upon which a virtual object (teapot) can be added.

method, is made of over-parameterized approaches which makes them easier to implement.

The linear solution, the easiest to compute, is the most unreliable because tensor is parameterized by all its entries and therefore does not take into account all the tensor geometrical constraints. The principle of both algebraic minimization and geometric distance minimization is to use the linear solution as an initial estimation and re-parameterize it by the 24 entries of the projection matrix $\mathbf{P}'$ and $\mathbf{P}''$. The desired tensor is found by minimizing the residual error. Because the tensor is estimated using all available matches, these methods can be affected by the presence of outliers. For this reason, there always is a risk of obtaining an invalid tensor.

The second class is the one that uses random sampling in the estimation of the tensor. The 6-point RANSAC (RANdom SAmple Consensus) method [9] has the capability of producing a good tensor estimate even in the presence of a significant number of mismatches [16]. The final tensor can then be estimated from the identified inlier matches using one of the aforementioned methods.

The main drawback of RANSAC is its computational complexity, which increases rapidly with the number of matches and proportion of outliers. Ways to improve the speed of RANSAC schemes are proposed in [12, 13] but a simpler approach will be used here. Another potential problem with RANSAC is that it does not always work well in a multi-planar scene. This problem has already been addressed in some papers [5, 18]. If the selected 6 points are accidently collinear or coplanar, the resulting tensor can only provide correct geometry for one plane. Consequently, to achieve the expected performances in both time and precision, we select algebraic minimization to estimate the tensor combined with the application of some additional steps to remove the inevitable outliers.

4.2. Modified Algebraic Minimization

The standard algorithm of algebraic minimization is briefly given here.

1. From the set of point triplets, compute the tensor linearly by solving a set of equations of the form $\mathbf{At} = \mathbf{0}$, where $\mathbf{A}$ expresses the equation (3) and $\mathbf{t}$ is the vector of entries of tensor.

2. Find the two epipoles $\mathbf{e}'$ and $\mathbf{e}''$ from the tensor

3. According to Equation (1), construct the 28×12 matrix $\mathbf{E}$ such that $\mathbf{t} = \mathbf{Ea}$, where $\mathbf{a}$ is the vector representing entries of $\mathbf{a_i}$ and $\mathbf{b_i}$.

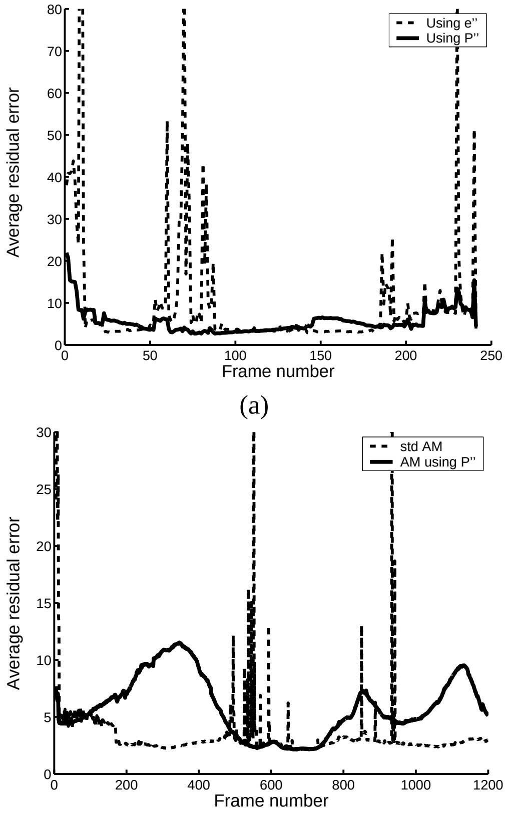
4. Compute the tensor by minimizing the algebraic error $||\mathbf{AEa}||$ subject to $||\mathbf{Ea}|| = \mathbf{1}$

In our framework, since the tensor is always associated with two fixed reference frames, one epipole and one projection matrix of this view triplet are actually fixed. Assuming that in Equation (1), $\mathbf{P_1}, \mathbf{P_3}$ correspond to the projection matrices of the two reference frames and $\mathbf{P_2}$ of the moving camera, then it follows that $\mathbf{P_3}$ is known from the initial tensor of the three reference frames and so is $\mathbf{e}''$. In Algebraic Minimization, the matrix $\mathbf{E}$ is constructed in terms of the 6 entries of the two epipoles, $\mathbf{e}'$ and $\mathbf{e}''$, which are retrieved from the tensor computed by the linear least-square solution. And the tensor is expressed linearly as $\mathbf{t} = \mathbf{Ea}$ where $\mathbf{a}$ is a vector of the entries of the left 3×3 matrices in $\mathbf{P_2}$ and $\mathbf{P_3}$. Now, to force the tensor to be consistent with the fixed $\mathbf{P_3}$ and $\mathbf{e}''$, two options are available: (1) choose the fixed $\mathbf{e}''$ instead of the one computed from the initial tensor of linear solution; (2) skip the step to solve the linear tensor and construct $\mathbf{E}$ directly with the entries of $\mathbf{P_3}$ and $\mathbf{a}$ with the entries of $\mathbf{P_2}$. In Figure 3 (a), we compare the performance of these two alternative approaches for a test sequence of 250 frames. From the experiment, it appears that the use of a fixed $\mathbf{e}''$ in the tensor computation is not very stable. In contrast, the tensor subject to a fixed $\mathbf{P_3}$ exhibits reliable performance over long sequences. It has the capability to counter the effect of false matches and comprise the conditions where few features are being tracked. However, its overall accuracy remains inferior to the one obtained with the tensor computed by standard Algebraic Minimization, as illustrated in Figure 3 (b). The instability of standard Algebraic Minimization when dealing with a small number of features results in peaks of error in the tensors estimated over a long sequence while the fixed $\mathbf{P_3}$ approach produces a much smoother curve. In consequence, the use of a fixed $\mathbf{P_3}$ will be restricted to the case where few features are available or a large portion of false matches exists or when tracked features are not well distributed across the whole scene.

Note that, in the estimation process, an affine tensor could have been used instead. The computation of this affine tensor is simpler and faster than the projective version, but it does not behave well for scenes with significant depth variations [11]. In addition, as it will be discussed in Section 7, the tensor computation does not constitute an important portion of the total processing time.

4.3. Establishing Point Triplets and Computing Trifocal Tensors

The task of tracking matched points belonging to one reference view is done using the widely used Lucas-Kanade tracker. During this tracking process, it is unavoidable that the tracker will lose some features, and will introduce some wrong traces. This is especially true in the case of sequences produced by handheld cameras involving quick and saccadic motion. Therefore, additional efforts are required in order to monitor the quality of tracked points and get rid of wrongly tracked points. However, it would be unreliable to measure the similarity between tracked points and their possible correspondences in the two reference images through normalized cross-correlation (NCC). Because it is observed that NCC degrades when there is a large rotation between images. An alternative solution is to apply a disparity-gradient constraint [19] on the candidate matches composed by the tracked points and their correspondences on the reference frames.



(a)

(b)

Figure 3. (a) comparison of the two modified Algebraic Minimization using the fixed $\mathbf{e}''$ or $\mathbf{P_3}$. The tensor computed using $\mathbf{e}''$ is drawn with a dash line. (b) comparison of the standard Algebraic Minimization and the one using the fixed $\mathbf{P_3}$ of a long sequence. The standard AM is fragile when few features are being tracked. The solid line represents the more stable tensor subject to $\mathbf{P_3}$

The resulting set of point triplets is used to compute a

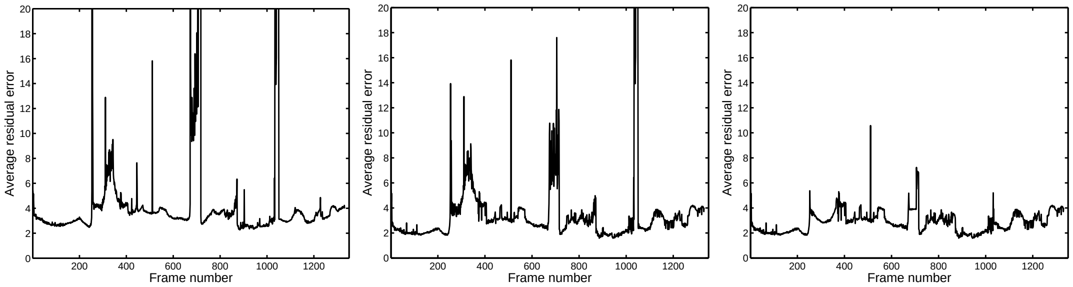


Figure 4. Experimental results for a sequence of 1331 frames. The left plot shows the residual error of a tensor computed with all putative triplets. The residual error of the new tensor computed after removing outliers is illustrated in the center plot. In the right plot, the remaining peak errors are eliminated by using a fixed P_3 in the tensor computation.

new tensor using the algebraic minimization approach. The average value of the residual errors is then used to assess the quality of the resulting transfer. If its value is smaller than a given threshold (we used 3 pixels), then the tensor is judged to be of good quality and can be used as is. Otherwise, additional steps to identify potential outliers are required; this is discussed next.

4.4. Removal of outliers

In the case where the quality of the tensor is lacking, it must be reestimated. The strategy used depends on the number of supporting triplets in the set of points. An important portion of supporting triplets means that the quality of the tensor is not good mainly because of the presence of a few strong outliers. In this case, a statistical method based on the so-called x84 rule [6] is implemented. Absolute deviations of all triplets' residual error are calculated, from which a threshold is automatically set as the 5.2 MAD(Median Absolute Deviation). The points having larger deviations are considered outliers and must then be eliminated.

In the opposite situation, i.e. when the number of supporting triplets is relatively low, then the current tensor is not able to guide the identification of outliers. Cross-correlation has to be performed on each putative triplet. All features on the current frame that do not correlate well their potential correspondences on both reference frames are rejected.

Once the outliers are rejected using one or the other of these methods, the tensor has to be re-estimated with all remaining triplets and its quality needs again to be re-evaluated. In Figure 4, the average residual errors of the tensors computed before and after the step of removing outliers are given.

5. Updating the Tracked Point Set

During tracking, the lifetime of a given tracked point largely depends on the magnitude of the camera motion. Over time, more and more points will thus be lost, mainly because they go out of the field of view or because they are occluded by some scene object. However, at the same time, other points, that are included in the initial pool of matches will appear in the view. It is therefore important, for the viability of the procedure, to identify those points and to incorporate them into the tracking process. Using the current estimation of the tensor, these points could be identified by transferring all the reference matches (in the two fixed views) onto the current moving view. The presence of a correspondence is verified by searching in a small area around the transferred point for a point that correlate well with one of the two reference matched points. This way, the method can even recover points that have been badly tracked in the current frame. The benefit of adding these points is illustrated in Figure 5.

6. Embedding Virtual Objects on the Transferred Pattern

Once the trifocal tensor of every video frame is obtained, rendering the virtual objects is accomplished in a two-step procedure:

1. The pattern from the reference images is transferred using the computed trifocal tensor.

2. The ARToolkit [10] is used to compute the homography from the *invisible* pattern plane to the XY plane of the virtual objects. A transformation matrix is then extracted from the homography and sent it to an openGL graphic server.

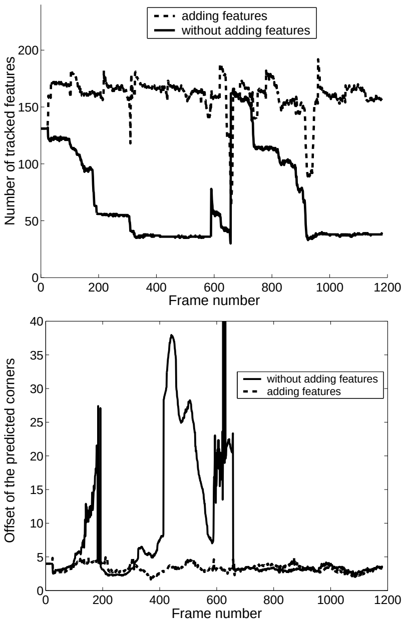


Figure 5. A long sequence(1183 frames) is recorded when a black-white square was placed on the desk. The camera's viewpoint has dramatic changes. Its four corners on every frame are detected and compared with the predicted corner positions computed from the estimated tensor. The number of tracked features across the sequence and the offset between tensor-predicted corners and their true detected positions before and after recovering lost and inaccurate features are shown respectively in the top and bottom plot. Note that in the without-adding-features case, the tracker is reset automatically when too few features remain. In this experiment, this happens at the $624th$ frame.

In our system it is not necessary to use a visible pattern, as is usually required by the ARToolkit. The same results are achieved by using an *invisible* plane transferred from the reference images by using the online computed tensor. This allows the continuing augmentation of the scene even if the pattern is removed after the initialization phase, or if it would be partly or completely occluded. A sample resulting video sequence is shown in Figure 6. A poster is augmented on the wall and is not lost, even though on some frames, the transferred pattern is almost out of view. Another benefit brought by the use of an *invisible* pattern is that virtual objects can be added anywhere, including on untextured surfaces (see Figure 7), as long as the surrounding regions have sufficient features.

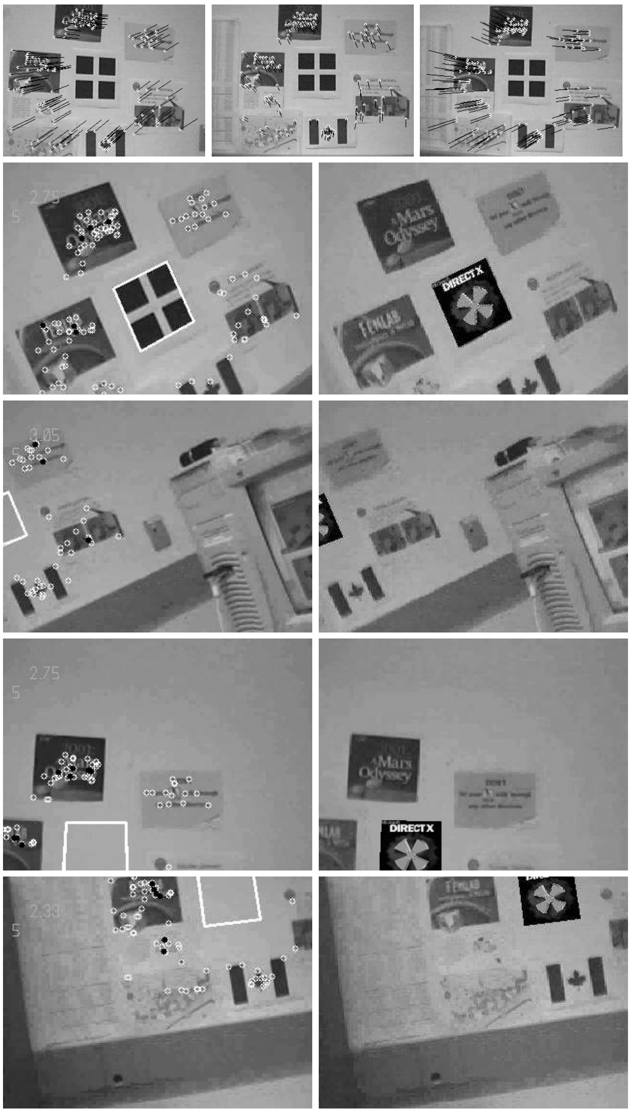


Figure 6. A square pattern is pasted on the wall when three reference frames are captured. The transferred patterns are shown superimposed on the video frames. From them, the homographies are computed which map a logo image on the wall.

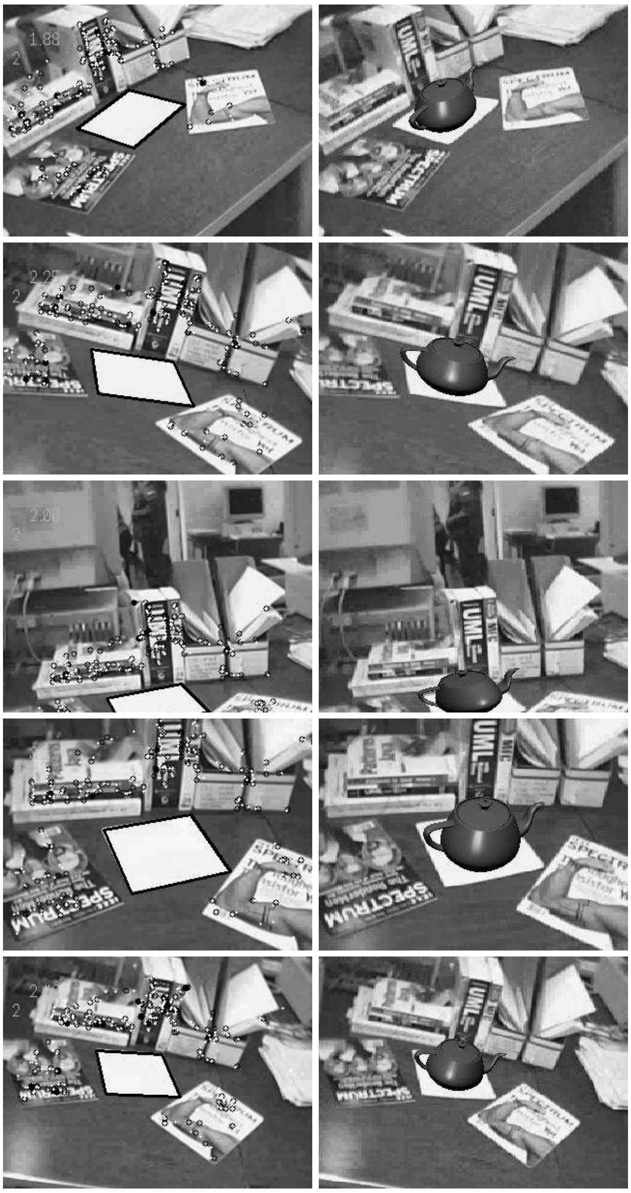


Figure 7. Example sequence: a teapot is added upon a white paper.

Chia [3] has used two keyframes to register the video frames of a calibrated camera. The approach described in this paper is an improvement in two respects. Firstly, In Chia's system, placement of virtual objects into the video relies on measurements of camera pose relative to two reference images, where positions of virtual objects have been computed by using the ARToolkit during initialization. We propose to implement an ARToolkit method with a transferred pattern along the video. Every frame is registered with respect to the graphic coordinate system individually. Therefore, the need for computation of camera pose and scene reconstruction is avoided. The online estimation of the trifocal tensor provides all the required information towards embedding virtual objects. Secondly, the trinocular geometry is exploited to provide a more powerful disambiguation constraint than the epipolar geometry would. This is because in a view triplet, image coordinates in a third view are completely determined, given a match in the other two view, whereas image positions are only restricted to a line by the epipolar geometry of image pairs.

7. Experimental Results

Our system runs on a desktop PC with a web camera of image resolution 320×240 pixels. The approach proposed in this paper has been tested on various sequences composed of thousand of frames each. Undoubtedly, its performance varies according to different conditions appearing in the videos. On average, the median residual error of the tensor is around 3 pixels and the processing is carried out at the speed of 14fps. An analysis of processing time is given in Figure 8.

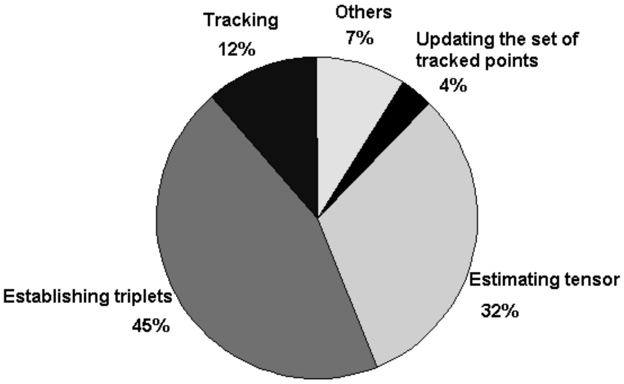


Figure 8. Timing chart

For each frame, about $8.32ms$, 12% of the time, is spent on tracking features along the sequence. Another 32% of the time is consumed on estimating the tensor. Establishing triplets for every video frame takes the largest portion of time. It varies with the number of features tracked along the sequence. Typically in our experiments the initial set of triplets over three reference images contains around 150 features.

Figure 7 shows some frames of an augmented video sequence. The frames with the inserted virtual object (the teapot) are shown on the right column, while the features points used to compute the tensors are identified on the left column. In this case, the pattern used for the augmentation corresponds to the white sheet that is shown enclosed by the reprojected black rectangle. More video sequences are available at the address: http://www.site.uottawa.ca/research/viva/projects/augmented/. Jitter of virtual objects on the video is observed when the camera looks at the edges of a scene containing multiple planes. One way to reduce this jittering would be to correct the homographies, using previous frames, in order to impose temporal smoothness of the transformations.

8. Conclusion

A new approach to augment a live video sequence was presented. It works in the context of a 3-view system, which consists of a moving camera and three reference images of the scene. The trinocular geometry relating every video frame to two of the reference images are estimated. Tensors are updated, online, over video stream, with a fast estimation method. This approach does not rely on camera pose to insert virtual objects, and does not require intrinsic camera calibration. The main requirements of the approach are that the scene must contain a sufficient number of features, and that these features must be well distributed. The proposed methodology works effectively as long as the moving camera captures views that remain within the visual hull spanned by the reference images. The performance of the system can be expanded by adding more reference views. We are currently investigating the use of an array of cameras where for each frame, the tensor between the two closest views will be estimated.

References

[1] http://www.hitl.washington.edu/artoolkit/.

[2] P. Beardsley, P. Torr, and A. Zisserman. 3d model acquisition from extended image sequences. In *European Conference on Computer Vision*, pages 683–695, 1996.

[3] K. Chia, A. Cheok, and S. Prince. Online 6 dof augmented reality registration from natural features. In *Proc. International Symposium on Mixed and Augmented Reality(ISMAR)*, 2002.

[4] K. Cornelis, M. Pollefeys, M. Vergauwen, and L. V. Gool. Augmented reality from uncalibrated video sequences. In *3D Structure from Images - SMILE 2000, Lecture Notes in Computer Science*, volume 2018, pages 144–160, 2001.

[5] F. Fraundorfer. *Robust Estimation of the Trifocal Tensor and it's Application to Image Matching*. PhD thesis, Graz University of Technology, 2001.

[6] A. Fusiello, E. Trucco, T. Tommasini, and V. Roberto. Improving feature tracking with robust statistics. In *Pattern Analysis and Applications*, volume 2, pages 312–320, 1999.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[8] K. Kutulakos and J. Vallino. Calibration-free augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 4:1–20, 1998.

[9] L.Quan. Invariants of six points and projective reconstruction from three uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, 17:34–46, 1995.

[10] S. Malik, C. McDonald, and G. Roth. Hand tracking for interactive pattern-based augmented reality. In *Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2002.

[11] P. Mendonca and R. Cipolla. Analysis and computation of an affine trifocal tensor. In *Proc. BMVC'98*, pages 125–133, 1998.

[12] D. Nister. Preemptive ransac for live structure and motion estimation. In *Proc. ICCV03*, pages 199–206, 2003.

[13] O.Chum and J.Matas. Randomized ransac with $t_{d,d}$ test. In *Proc. BMVC'02*, pages 448–457, 2002.

[14] G. Roth and A. Whitehead. Using projective vision to find camera positions in an image sequence. In *Vision Interface*, 2000.

[15] Z. Sun and A. Tekalp. Trifocal motion modeling for object-based video compression and manipulation. *IEEE Transactions on circuits and system for video technology*, 8, 1998.

[16] P. Torr and A. Zimmerman. Robust parametrization and computation of the trifocal tensor. *Image and Vision Computing*, 15, 1997.

[17] L. Vacchetti, V. Lepetit, and P. Fua. Fusing online and offline information for stable 3d tracking in real-time. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2003.

[18] J. Vigueras-Gomez, M. Berger, and G. Simon. Iterative multi-planar camera calibration : Improving stability using model selection. In *Vision, Video and Graphics(VVG)'03*, 2003.

[19] E. Vincent and R. Laganière. Matching feature points in stereo pairs: A comparative study of some matching strategies. *Machine Graphics & Vision*, 10:237–259, 2001.

[20] E. Vincent and R. Laganière. Matching feature points for telerobotics. In *Proc. 1st International Workshop on Haptic Audio Video Environments and their Applications*, pages 13–18, 2002.

[21] A. Zisserman, A. Fitzgibbon, and G. Cross. Vhs to vrml: 3d graphical models from video sequences. In *IEEE International Conference on Multimedia and Systems*, volume 1, pages 51–57, 1999.